

Application Serial Number 10/038,547
Attorney Docket Number RA-5368

Response Filed 6/16/2005

Claim Amendments

Please amend the Claims as follows:

Claim 1 (Cancelled)

Claim 2 (Cancelled)

Claim 3 (Cancelled)

Claim 4 (Cancelled)

1 5. (Currently Amended). A second stage lottery program for a dispatcher program
2 that dispatches tasks within an operating system of a computer system, the computer
3 system supporting at least two classes of said tasks, each of said classes including at
4 least two levels of said tasks, said dispatcher program to determine which of said tasks
5 ~~wherein said second stage lottery system determines which of at least two tasks of at~~
6 ~~least two levels within a selected class of at least two task type classes~~ will be assigned
7 to a next available IP resource ~~available to a scheduler queue on which pointers to said~~
8 ~~at least two tasks reside~~, said second stage lottery program comprising:
9 a random number generator and selection program for generating a first random
10 number and for selecting one of said at least two classes, and for generating a
11 second random number for selecting one of said at least two levels within said
12 selected class; ~~based upon a correspondence of a thereby generated random~~
13 ~~number and~~
14 a transfer program for transferring control of said next available IP resource from
15 said second stage lottery program to a task found on assigned to said selected one
16 of said at least two levels.

Application Serial Number 10/038,547
Attorney Docket Number RA-5368

Response Filed 6/16/2005

1 6. (Currently Amended) The second stage lottery program of claim 5 further
2 comprising:

3 a level switching routine for handling a failure by said transfer program to find
4 a task on a said selected one of said at least two levels, said level switching routine
5 for selecting a different one of said at least two levels that is associated with a task,
6 and modifying said level selecting of a one of said at least two levels by said random
7 number generator and selection program to a second one of said at least two levels,
8 iteratively, until a task is found on a selected level, and when a task is found,
9 allowing transfer of control to said ~~found~~ task associated with said different one of
10 said at least two levels.

1 7. (Currently Amended) The second stage lottery program of claim 5 wherein any
2 one of said at least two levels ~~selected among by said random number generator and~~
3 ~~selection program are constructed wherein each next higher level among said at least~~
4 ~~two levels~~ is two times more likely to be selected than a next lower one of its next lower
5 ~~level among~~ said at least two levels.

1 8. (Currently Amended) The second stage lottery program of claim 7 wherein each
2 level of said at least two levels will ~~only~~ have tasks of like quantum values, ~~within said~~
3 ~~each level~~ wherein each of said quantum values defines an amount of time a task will
4 be assigned to said next available IP resource.

1 9 (Currently Amended). The second stage lottery program of claim 5 wherein each of
2 said tasks has a quantum value, said quantum value identifying a computer system
3 specific amount of time in which said each of said tasks with said quantum value may
4 continuously execute on an instruction processor resource, and wherein said second
5 stage lottery employs a quantum bias routine, said quantum bias routine comprising:
6 a data capture routine for determining how much of an the allotted segment of a said

Application Serial Number 10/038,547
Attorney Docket Number RA-5368

Response Filed 6/16/2005

7 quantum value a task that has executed used before returning control to said
8 dispatcher[.]; and
9 a bias adjustment routine for adjusting ~~a current~~ said allotted segment of a said
10 quantum value based on how much of said allotted segment was used ~~identifying~~
11 ~~how much of said quantum said task that has executed has to a new allotted~~
12 ~~segment of a quantum for said a task.~~

1 10 (Currently Amended). The second stage lottery program of claim 9 wherein said
2 bias adjustment routine does not adjust said current allotted segment to a new allotted
3 segment for said task ~~that has executed if said task that has executed's use of its~~
4 ~~current~~ if use of said allotted segment was interrupted by an interrupt.

1 11. (Canceled)

1 12. (Currently Amended) A computer system having a quantum timer settable to allow
2 processing on an IP resource for a limited duration by any task of one or more tasks,
3 also having an operating system having a dispatcher program wherein all said any tasks
4 are identifiable as being members of classes and wherein said dispatcher program
5 comprises:

6 a) A a scheduler code section executable to determine for how long, and to
7 which of said any one or more tasks, ~~pending in a scheduler queue that an IP~~
8 ~~resource will be next assigned to process on said IP resource and for how long~~
9 ~~said next assigned task may process on said IP resource;~~ and

10 b) A a scheduler queue from which said any one or more tasks may be
11 addressable and assignable to said IP resource, wherein

12 said scheduler code section has a two stage lottery execution algorithm, a
13 first of said two stages using a lottery process for determining from which class of
14 said classes a next one of said any one or more tasks will be selected ~~by a~~
15 ~~second stage lottery execution algorithm to process on said IP resource, said~~

Application Serial Number 10/038,547
Attorney Docket Number RA-5368

Response Filed 6/16/2005

16 second of said two stages using a second lottery process to select determining
17 ~~which among said any tasks that happen to be within said class will be said next~~
18 assigned a task that will next be assigned to said IP resource from a class
19 determined by said first of said two stages.

1 13. (Currently Amended) The computer system set forth in claim 12 wherein said first
2 stage ~~lottery execution algorithm performs to choose~~ chooses a class randomly ~~but with~~
3 ~~a bias settable by a user~~ from among all said classes using a bias settable by a user.

1 14. (Currently Amended) The computer system set forth in claim 12 wherein ~~athe~~
2 number ~~determining how many there are of~~ said classes is selectable by a user.

1 15. (Currently Amended) The computer system as set forth in claim 12, wherein if any
2 one of said any tasks is of a class "above the lottery line" then said any one of said any
3 tasks is assigned an IP resource prior to running said scheduler code section.

1 16. (Original) The computer system as set forth in claim 12, wherein if the scheduler
2 code section's first of two stages selects a class which is empty of said any tasks, said
3 scheduler code section next chooses another class of available classes.

1 17. (Original) The computer system as set forth in claim 12, wherein if none of the
2 classes has any of said any tasks, said scheduler code selects a very low priority level
3 operating system task.

1 18. (Currently Amended) The computer system as set forth in claim 12, wherein said
2 second stage lottery comprises:

Application Serial Number 10/038,547
Attorney Docket Number RA-5368

Response Filed 6/16/2005

3 a random number generator and selection program for generating a random number
4 and for selecting a one of said at least two levels within said selected class based
5 upon a correspondence of a said thereby generated random number[.]; and
6 a transfer program for transferring control from said second stage lottery program to
7 a task found on said selected one of said at least two levels.

1 19. (Currently Amended) A method for use by a dispatcher algorithm in an operating
2 system in a computer system for selecting a task to provide with an available instruction
3 processor resource, wherein the task may be selected from one or more classes, each
4 of which may have one or more priority levels, wherein said method comprises:
5 determining whether within a scheduler queue there are any task pointers
6 within said priority levels of a class said classes of tasks on said scheduler
7 queue, and if so, determining whether any said any task pointers are above a
8 second stage lottery line, and if so assigning a first of said any tasks indicated by
9 said any task pointers above said second stage lottery line to said available
10 instruction processor resource[.]; but if not,
11 and if there is only one said priority level having any of said any task
12 pointers, assigning a first of said any tasks corresponding to said any task
13 pointers at said only one said priority level to said available instruction processor
14 resource[.]; else,
15 running a first stage of a second stage lottery algorithm to select one of
16 said classes, and running a second stage of said second stage lottery algorithm
17 to select from said one of said classes a priority level from which to select said
18 task to provide to said available instruction processor resource ~~determine to~~
19 ~~which priority level among a plurality of said priority levels having said any task~~
20 ~~pointers said available instruction processor resource should be made available~~
21 ~~to assign to a task within said which priority level.~~

Application Serial Number 10/038,547
Attorney Docket Number RA-5368

Response Filed 6/16/2005

1 20. (Currently Amended) A method for use by a dispatcher algorithm in an operating
2 system in a computer system for selecting a task from one or more classes to provide
3 with an available instruction processor resource wherein said method comprises:

4 determining whether within a scheduler queue there are any task pointers
5 within priority levels of a class of tasks on said scheduler queue, and if so, and if
6 there is only one said priority level having any of said any task pointers,
7 assigning a first of said any tasks corresponding to said any task pointers at said
8 only one said priority level to said available instruction processor resource, else,

9 running a second stage lottery algorithm, a first lottery stage of said lottery
10 algorithm to select one of the plurality of classes, and a second lottery stage of
11 said lottery algorithm to select from said one of said classes a to-determine-to
12 which priority level among a plurality of said priority levels having said any task
13 pointers from which to select a task to assign to said available instruction
14 processor resource should be made available to assign to a task within said
15 which priority level.

1 21. (Currently Amended) The method of claim 20 further comprising:

2 moving said task pointers within priority levels of said one or more
3 classes ~~a class of tasks~~ on said scheduler queue, wherein each priority level can
4 maintain a chain of said task pointers and wherein there are more than one of
5 said priority levels, said task pointer moving process comprising:

6 maintaining a task assigned quantum which identifies for each task on
7 said scheduler queue a set amount of instruction processor resource to which
8 said each task is entitled upon being assigned to ~~an~~ said instruction processor
9 resource[.];

10 placing said each task into a said priority level based upon a value in said
11 task assigned quantum for said each task[.]; and

12 changing said each task priority level based on how much of said value in

Application Serial Number 10/038,547
Attorney Docket Number RA-5368

Response Filed 6/16/2005

13 said task assigned quantum for said each task said each task used a last time
14 said each task was assigned to an said available instruction processor resource.

1 22. (Currently Amended) The method of claim 21, and further comprising: wherein
2 an algorithm for determining whether said each task priority level should be changed
3 based on said last time said each task was assigned to an said available instruction
4 processor resource is: whereby

5 _____ if said last time did not complete within said task assigned quantum, lower said
6 task priority level for said this task by one priority level and raise said value of said task
7 assigned quantum by a factor of 2,

8 _____ if said last time used less than said value of said task assigned quantum but
9 more than a small portion of said task assigned quantum, leave said priority level and
10 said small portion the same, and

11 if said last time used less than said small portion of said value of said task
12 assigned quantum, increase ~~this task's~~ said priority level of said task by one and halve
13 ~~said this task's~~ task assigned quantum value of said task.

1 23. (Currently Amended) The method of claim 21 wherein said task assigned
2 quantum ~~portion~~ is assigned for said each task prior to a said task pointer being on said
3 scheduler queue.

1 24. (Original) The method of claim 21 wherein a first stage lottery algorithm selects
2 which of said classes of tasks on said scheduler queue will have tasks assignable to
3 said available instruction processor resource.

1 25. (Original) The method of claim 21 wherein a first stage lottery algorithm selects
2 which of said classes of tasks on said scheduler queue will have tasks assignable to

Application Serial Number 10/038,547
Attorney Docket Number RA-5368

Response Filed 6/16/2005

3 said available instruction processor resource, but only if there are no tasks in classes
4 above a first lottery line, and if there are tasks in said classes above said first lottery
5 line, providing said available instruction processor resource to said tasks in classes
6 above said first lottery line in a priority order.

1 26. (Currently Amended) A dispatcher system for use within a computer system, said
2 computer system having an available instruction processor resource and a scheduler for
3 storing pointers to tasks to be assigned quantum of said instruction processor
4 resource, said dispatcher system comprising:
5 a first stage of a two stage lottery system for using a random number generator
6 to select one of multiple classes of tasks; and
7 a second stage of said two stage lottery system for selecting a task from said
8 selected one of said multiple classes, whereby through which each said selected task is
9 assigned to an available instruction processor resource for a portion of a quantum
10 assigned to said selected task.